Agile e6 — Upgrade tool 3.0

Overview Upgrade Process

Copyrights and Trademarks

July 7, 2005

CONTENTS

Chapter 1
Introduction

This guide is intended as a manual for upgrading earlier versions of the Agile e-series products to Agile e6.0 with the Agile Upgrade Tool.

The Upgrade tool allows a direct upgrade to Agile e6.0 from one of the following product versions

- ❑ CADIM/EDB 2.3.2 or higher
- ❑ axalant 2000 SPx
- ❑ Eigner PLM 5.x

Additional Information

In addition to this document, the following documents are available in the **upgrade/doc/** directory:

- ❑ Installation and Configuration manual
  Describing how the Upgrade Tool is installed and configured before the migration to Agile e6.
- ❑ Migration manual
  Describing the actual migration from your existing product version to Agile e6.
- ❑ Annex
  Listing additional information e.g. directory structure, Shell and SQL scripts, Configuration parameters etc.
- ❑ Release Notes
  Describing changes between the old Upgrade Tool and the Upgrade Tool 3.0
- ❑ Dump Comparison
  Listing all dump changes made from Agile e5.1 and Agile e6.0

For additional information and most up-to-date Upgrade information, check the Agile Support page at http://eignersupport.agilesoft.com/index.asp (you will need a password to enter the support website).

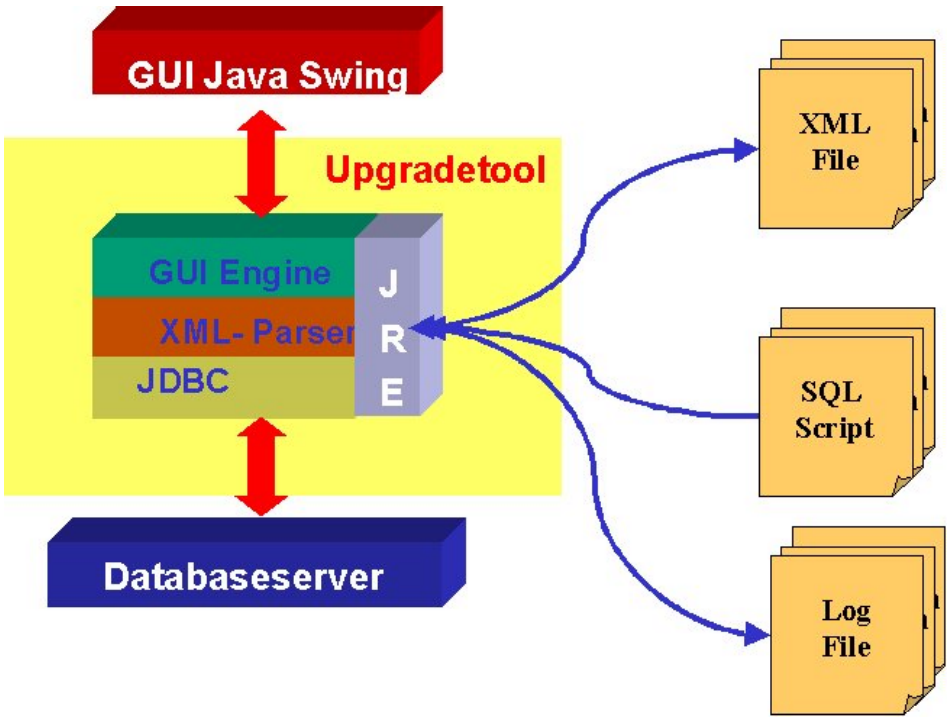
Chapter 2

Architecture and Basic Concepts

Architecture

The upgrade tool is implemented in Java. The tool accesses the databases directly using a JDBC connection. The configuration of all upgrade steps is stored in a set of xml control files. In addition SQL scripts are used for special steps. The log and error files are mainly in XML format.

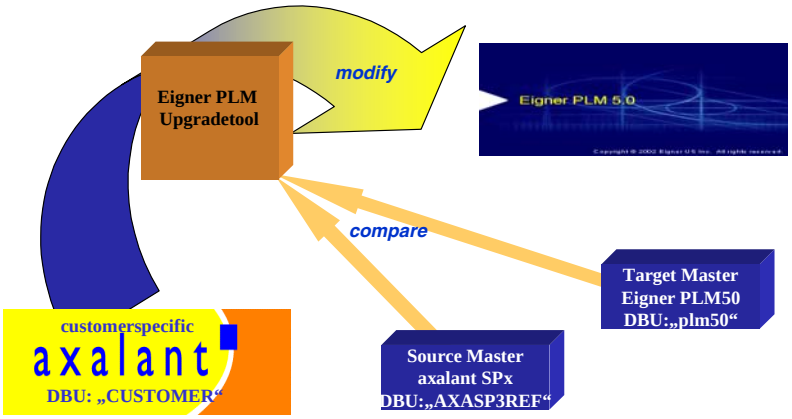
The user can execute the migration steps easily from the user interface of the Upgrade Tool.



Concept Customization Upgrade

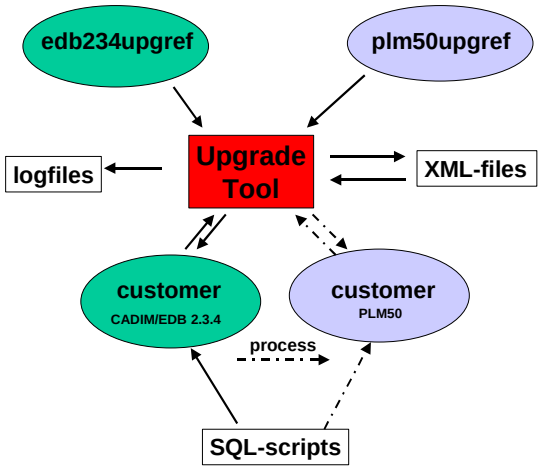
The Upgrade Tool opens three database connections:

- Source Master (*Source reference*):
  A CADIM/EDB / axalant 2000 / Eigner PLM dump with Agile e6.0 ”n” table structure. The necessary SQL scripts have been executed to adjust the DataView tables. LogiView standard models have been deleted so that standard LogiView models will be completely reloaded. This is already done in reference dumps delivered with this upgrade tool.



- Target Master (*PLM reference*):
  This is an Agile e6.0 dump.
- Customer (*CUSTOMER*):

  The customer dump has the version of the Source Master at the beginning of the upgrade process. At the end it has the version of the Target Master.



The Upgrade Tool selects each row from the Source and the Target Master dump, compares the data sets from both dumps to identify the differences and checks if the customer has modified these data.

The upgrade action (Insert, Update, Delete) is determined for each record and the information is stored in a set of XML files. The migration rules are listed in the Annex.

The Upgrade Tool is only able to compare tables with the same table structure. Therefore the DataView tables in the reference dumps (edb234upgref …) have Agile e6.0 structure. The customer dump will be formatted during the upgrade (execution of SQL scripts and step synchronize repository).

The customization upgrade is generally split into two steps:

Comparison of the data sets from the different dumps and storing the changes in an XML file for the three possible operations: delete, insert and upgrade (e.g.: dtvdel.xml, dtvins.xml, dtvupg.xml).

The Upgrade Tool reads the XML information and performs the corresponding SQL-statements. After this step the changes are available in the Agile e6.0 dump.

Concept Transfer Data from productive system

After finishing the test of the new custom-specific Agile e6.0 functionality, the customer can go live with the new version.

During the test period a lot of new data is created in the productive system. This data must be copied again from the old productive system into the new environment and adapted to the new Agile e6.0 table structure. This step is called Takeover Productive data.

The Upgrade Tool supports the following actions:

- Definition of a list of reference tables containing productive data.
- Dropping the reference tables in the customer environment and copy the table from the productive system.
- Adapting the table structure to Agile e6.0.
- Post-actions like migrate productive classification data.

For this step the tool opens two database connections:

- Productive Database

  This database connection is used as source for the tables copied into the new Agile e6.0 environment. No changes are made in the productive database.

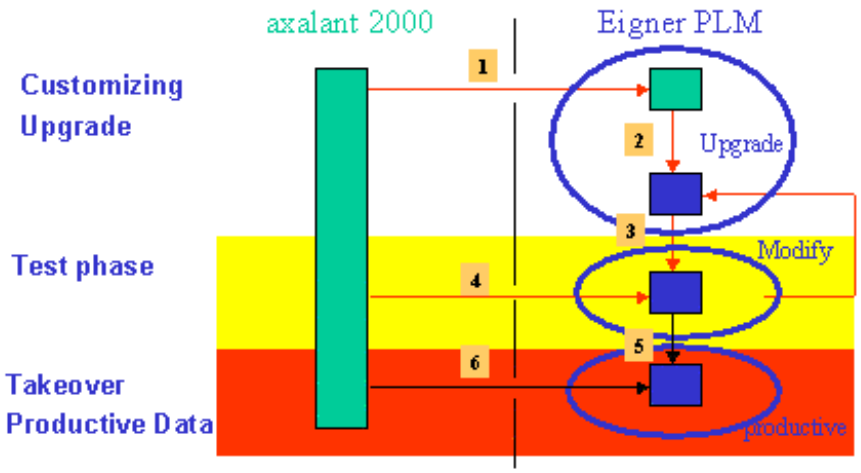
- Customer

  Connection to the new production environment. The reference tables will be dropped. This becomes the new productive Agile e6.0 environment.

Chapter 3

Overview Upgrade Process

The Upgrade tool supports the following actions of your migration process to Agile e6.0



- Preactions on the original production environment
- Customizing Upgrade
  In this step the customization and configuration stored in the database is updated to Agile e6.0.
- Test phase
  After this step you must test if all functionalities run correctly, including your own customer-specific functionality.
- Take over productive data
  All tables containing productive data like document and item master data are copied from the production system to your Agile e6.0 installation. They will be adapted so that you can work on this data within Agile e6.0.

The migration process is carried out in the following steps:

Pre-action activities on the original productive environment
Make a copy of your productive database dump. Do not attach your productive system. Always work on a copy of your data.

Start the upgrade on this copy. The minimum passing time will be 4-5 hours (depending on the system, main parameter is memory!).

The Upgrade Tool will create a dump on which you can run Agile e6.0. This dump is not error free. You have to check all functionalities and clear out the errors caused by setting up the upgrade tool.

Take over the data from the production system for a first test.

Let the user test all functionalities, maybe during training. If errors occur, remove them via customizing. Not everything might be done automatically.

If testing does not raise any error you can plan to change to productivity. Shut down your CADIM/EDB, axalant or Eigner PLM system. Take over your data from the production system (and the files!) again and Agile e6.0 is your productive system.

> Note: You should always be able to restart the CADIM/EDB, axalant or Eigner PLM system as a fallback strategy.

Chapter 4

Special Concepts for Upgrade Steps

Concept Classification Upgrade

If you use the ATT Concept for classification special upgrade steps are necessary

Depending on your source version you have to execute the following steps:

- Migration of classification data from the ATT concept to the new pool concept

  This step is necessary for migration from CADIM and axalant to Agile 6.0
- Migration of changes in the attribute inheritance

  Must be executed for all version

Overview ATT concept (Old)

- Attributes are defined class specific in the ATT concept
- Domain values for an attribute are defined in static menus
- No release procedures and status management for classes and attributes

Overview new Pool concept

- Attributes can be defined class independed
- Pool attributs can be assigned to more then one class
- Domain values for a pool attribute can be stored in special domain table
- For every class can be specified which domain value is valid

Migration concept

The migration includes

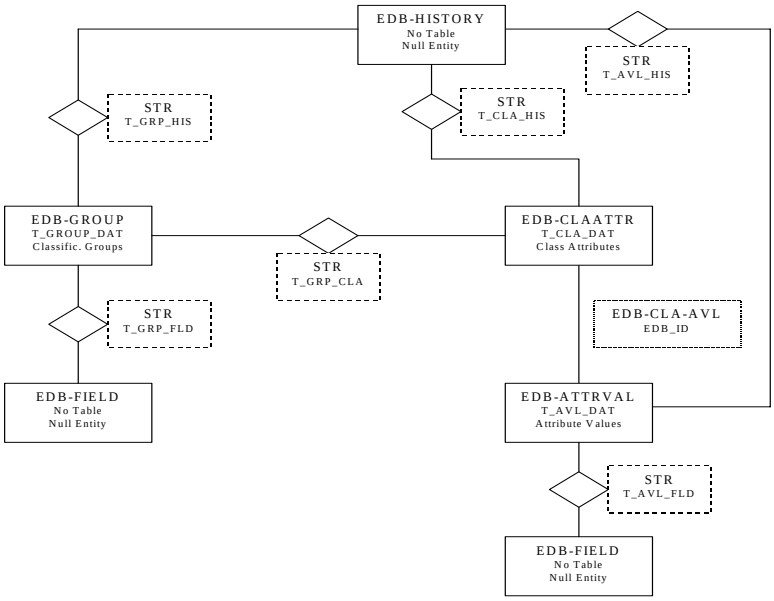
- Merge of Attribute definition

  Attributes are considered as identical if the following values are identical

  - C_LETTER
  - C_TITLE
  - C_TYPE

  If you have defined C-Letter and C_TITLE as multi language fields(Standard beginning with axalant 2000) you define with the Parameter DB language which language is used as basic for the merge. (see installation and configuration manual for more information how to define parameters)
- Initial Load of the attribute value pool including the activation of the attributes for special classes

- ❑ Update classification lists
- ❑ Update used field name
- ❑ Set of attribute ATT_VAL_REF in the classification Lists
- ❑ Update field and Mask definition (if you have defined own forms for classes



If possible do not create or modify basic definition of classes and attributes between customization upgrade and take over data from production system.

This influences the migration steps must be executed after the takeover process.

- ❑ No new classes and attributes are created

  Only classification list tables must be defined as reference table

  - T_GRP_ART
  - T_GRP_DOC
  - T_GRP_ORD
  - T_GRP_PRO

- ❑ Customers have created new classes and attributes in the production system after the customization upgrade

  In addition to the classification list tables classes, attributes and domain values must be copied and migrated

  - T_GROUP_DAT
  - T_GROUP_STR
  - T_GRP_FLD

Pre-action activities on the original productive environment

Since Agile e5.0 UIC and GIC <=1000 are reserved for the standard development. Existing users or groups use such C_IC must be migrated to an higher Value.

This migration must be executed in your production system before you start any other upgrade activity.

This update can be very time consuming ! In a big customer dump it takes 1h/6 users. To solve the time conflict, breakdown the update into different sub sets and try to run it in parallel .

1. To do this Adapt the following statement in the sql script

```
INSERT INTO TEMP_U (OLD_U) SELECT C_IC FROM T_USER a
WHERE
C_IC > 200 AND C_IC < 1000 AND C_NAME NOT LIKE 'EDB%' AND
C_NAME NOT LIKE 'DEMOEP%';
```

To execute the UIC/GIC Migration execute the

```
PC          RUN update_uic.cmd
UNIX                run update_uic.sh
```

Take Over Data from the productive system

If you created new users and or groups since the date when the dump was exported form productive system the following DataView-tables must be migrated !

- T_USER,
- T_GROUP,
- T_GRP_USR,
- T_PROFILE
- Related tables of the PLM – person management

Attention: new plm- or axalant-user like EDB-WFL, EDB-DFM, EDB-DDM, EDB-GDM, EDB-EER, DODEKERNEL will be lost. Export these users first with the binary loader (T_USER, T_GROUP, T_GRP_USR) and reload them after the upgrade. Table T_DEFAULT should be migrated by the loader (import/overload) otherwise new defaults will be missing.